What is claimed is:

1. An electronic control unit having implemented on it, and made up of components, a software in which a plurality of software interfaces, optimized with respect to the exchange of data, is provided for the optional coupling in of a plurality of applications, the software including at least one application-specific software code of a component for each application capable of being coupled in, which is activated when the application is coupled in.

2. The electronic control unit as recited in Claim 1,
in which the software has a hierarchical layer architecture with respect to a mutual access possibility of the software components, each software component being assigned to a layer.

3. The electronic control unit as recited in Claim 2,
in which the layer architecture implements a separation of hardware-independent software components from hardware-dependent software components.

4. The electronic control unit as recited in one of Claims 1, 2, or 3,
in which the software interfaces are grouped into „on board" and "off board" interfaces as a function of the respective applications that may be coupled to them.

5. The electronic control unit as recited in one of the preceding claims,
in which the software assumes in each case one of several possible operating states, as a function of data present at the software interfaces and therein supports exclusively condition-specific functionalities.

6. The electronic control unit as recited in one of the preceding claims,
in which the software may assume, as the operating state, a "software update" state, a "software parameterization" state, a "software diagnosis" state, a "coasting" state, a "monitoring" state or an "on" state.

7. The use of an electronic control unit (100) as recited in one of the preceding claims, in an automobile electronic system, especially for controlling operating sequences in a vehicle.

8. A method for specifying a software architecture for an electronic control unit (100), in which after the specification of defined software interfaces, of software components, of software layers and of software operating states, automatically in each case an assignment of

- the software components to the software layers and to the software operating states is undertaken, the assignments being verified by subsequent analysis and checking of the interactions implemented based on the assignments.

9. The method as recited in Claim 8,
in which, in the case of insufficient assignments, the software components are subdivided into specified subcomponents and/or the software layers into specified sublayers and/or the software operating states into specified substates, and a renewed assignment is carried out automatically.

10. A computer program having program code means, by which a method as recited in one of Claims 8 or 9 is automatically carried out when the program code elements are executed on a computer or on a computer system.

11. A computer program product having program code means, which are stored on a computer-readable data carrier, in order to carry out a method according to one of Claims 8 or 9 when the computer program is executed on a computer.